

AJAX example with useEffect, .env.local etc.

### --- index.html (unchanged) ---

```
<body>
  <noscript>You need to enable JavaScript to run this app.</noscript>
  <div id="root"></div>
  <!--
    This HTML file is a template.
    If you open it directly in the browser, you will see an empty page.

    You can add webfonts, meta tags, or analytics to this file.
    The build step will place the bundled scripts into the <body> tag.

    To begin the development, run `npm start` or `yarn start`.
    To create a production bundle, use `npm run build` or `yarn build`.
  -->
</body>
```

### --- index.js (unchanged)---

...

```
const root = ReactDOM.createRoot(document.getElementById('root'));
root.render(
  <React.StrictMode>
    <App />
  </React.StrictMode>
);
```

### ---App.js ---

```
import './App.css';
import CategoryListView from './views/CategoryListView';

function App() {
  return (
    <div className="App">
      <CategoryListView />
    </div>
  );
}
```

```
export default App;
```

### ---.env.local ---

```
REACT_APP_BE_SERVER_BASE_URL=http://localhost:8777/api
```

```
--- dao.js ---
const baseUrl = process.env.REACT_APP_BE_SERVER_BASE_URL;

const fetchCategories = async () => {
  const resp = await fetch(`${baseUrl}/category/all`);
  const data = await resp.json();
  return data;
};

const dao = {
  fetchCategories,
};

export default dao;
--- CategoryRow.js ---

import React from "react";

const CategoryRow = ({index, category}) => {
  return (<div>{category.id + ":" +category.name}</div>);
}

export default CategoryRow;
```

### --- CategoryListView.js ---

```
import React, {useState, useEffect} from "react";
import CategoryRow from "../components/CategoryRow";
import dao from "../ajax/dao";

const CategoryListView = () => {
  const [categories, setCategories] = useState([]);

  useEffect(() => {
    // ...do something ONLY when component did mount
    // Notice: even if categories state change => not fired
    async function refreshCategories () {
      const data = await dao.fetchCategories();
      setCategories(data);
    }
    refreshCategories();
  }, []);

  useEffect(() => {
    // ...do something when 1. component did mount
    // 2. categories state changes
  }, [categories]);

  return (
    <>
    <h3>Categories</h3>
    <div>
      {categories && categories.length>0 ?
        categories.map((item)=>
          <CategoryRow      key={item.id}
                           id={item.id}
                           category={item} />
        ) : "no categories yet"
      }
    </div>
    </>
  );
};
```